**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 5&6**

**I/O, TIPE DATA & VARIABEL**

**Sebuah gambar berisi logo, teks, simbol, Grafis

Konten yang dihasilkan AI mungkin salah.**

**Disusun oleh:**

**RIZKY TABRIZ DEANOVA**

**109082500177**

**S1IF-13-07**

**Asisten Praktikum**

Adithana Dharma Putra

Apri Pandu Wicaksono

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

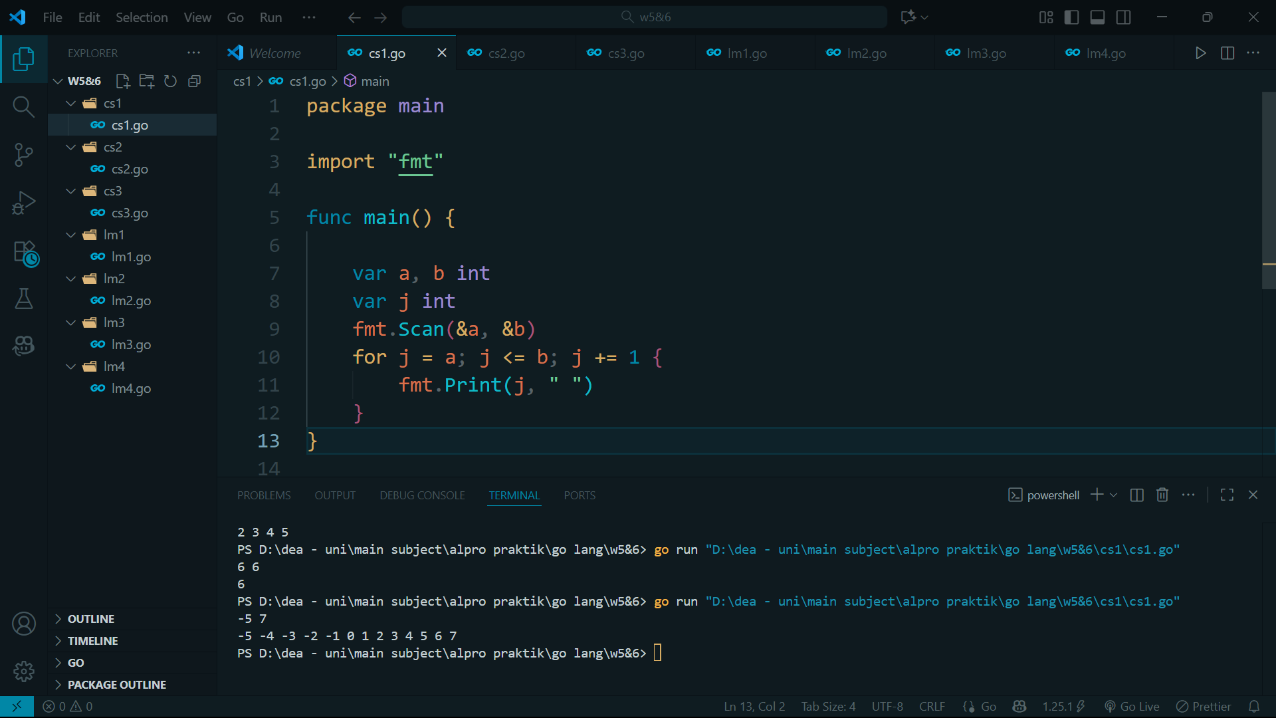
**LATIHAN KELAS – GUIDED**

1. **Guided 1**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var a, b int      var j int      fmt.Scan(&a, &b)      for j = a; j <= b; j += 1 {          fmt.Print(j, " ")      }  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Print yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Cetak Nilai** yang mana terlihat jelas melalui kode dari **baris 10 hingga baris 11**.

for j = a; j <= b; j += 1 🡪 Memberikan instruksi untuk melakukan perulangan variabel j yang dimulai dari nilai variabel a hingga nilai variabel b, selama variabel j tidak lebih besar dari nilai variabel b (j <= b), maka akan dilakukan perulangan pertambahan 1 nilai untuk output nilai sebelumnya di setiap perulangan j (j += 1), hingga mencapai ketentuan akhir, yaitu nilai variabel j kurang dari sama dengan nilai variabel b (j <= b).

fmt.Print(j, “ ”) 🡪 Instruksi untuk mencetak output dari variabel j atau instruksi untuk mencetak output dari perulangan variabel j dari baris kode sebelumnya.

Runtutan Eksekusi:

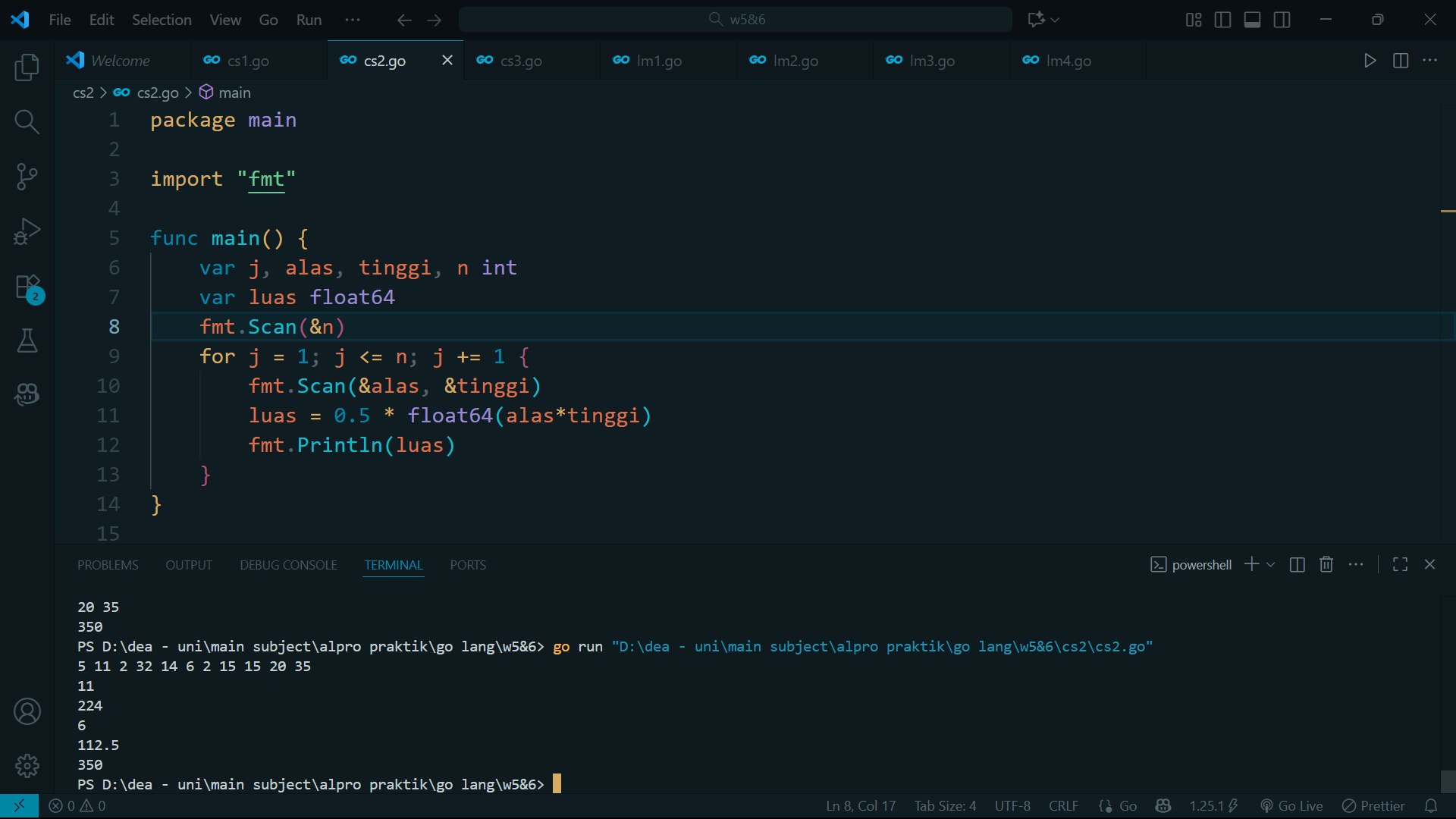
* + - 1. Menunggu user melakukan input data (fmt.Scan(&a, &b)) yang nantinya data tersebut akan dimasukkan ke dalam variabel a dan b. Hasil dari kode di baris ini adalah -5 7
      2. Melakukan perulangan variabel j menggunakan nilai dari variabel a dan variabel b (for j = a; j <= b; j += 1), yang mana untuk setiap nilai variabel j yang kurang dari variabel b maka akan ditambahkan 1 hingga output perulangan tersebut mencapai nilai variabel j yang sama dengan nilai variabel b. Hasil dari kode di baris ini adalah perulangan -5+1 = -4, -4+1 = -3 hingga 6+1 = 7, yang mana nilai 7 sama dengan nilai variabel b dan perulangan akan berhenti di nilai yang sama dengan nilai variabel b
      3. Melakukan print output dari variabel j (fmt.Print(j, " ")), yang mana output dari variabel j didapatkan dari perulangan di kode baris sebelumnya. Hasil dari kode di baris ini adalah -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7

1. **Guided 2**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var j, alas, tinggi, n int      var luas float64      fmt.Scan(&n)      for j = 1; j <= n; j += 1 {          fmt.Scan(&alas, &tinggi)          luas = 0.5 \* float64(alas\*tinggi)          fmt.Println(luas)      }  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Luas Segitiga Sebanyak Nilai N** yang mana terlihat jelas melalui kode dari **baris 9 hingga baris 12**.

for j = 1; j <= n; j += 1 {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for j = 1; j <= n; j += 1

fmt.Scan(&alas, &tinggi) 🡪 Instruksi untuk memasukkan 2 input nilai oleh user ke dalam variabel alas dan variabel tinggi, sesuai urutan nilai yang diinput oleh user. Dalam kode baris ini, setiap 2 nilai yang diinput oleh user untuk masuk ke dalam data variabel alas dan variabel tinggi terhitung 1 nilainya (j += 1). Oleh karena itu, perulangan input nilai dilakukan berulang hingga 5 kali, yang mana nilai 5 merupakan nilai dari variabel n, perulangan input nilai akan berhenti hanya jika jumlah input nilai sudah terhitung sama dengan 5 (j <= n).

luas = 0.5 \* float64(alat\*tinggi) 🡪 Memberikan instruksi untuk melakukan operasi hitung luas dari setiap nilai dalam variabel alas dan tinggi.

fmt.Println(luas) 🡪 Memberikan instruksi untuk melakukan pencetakan hasil dari setiap operasi hitung luas untuk seluruh nilai dalam variabel alas dan tinggi dari operasi hitung dalam kode baris sebelumnya.

Runtutan Eksekusi:

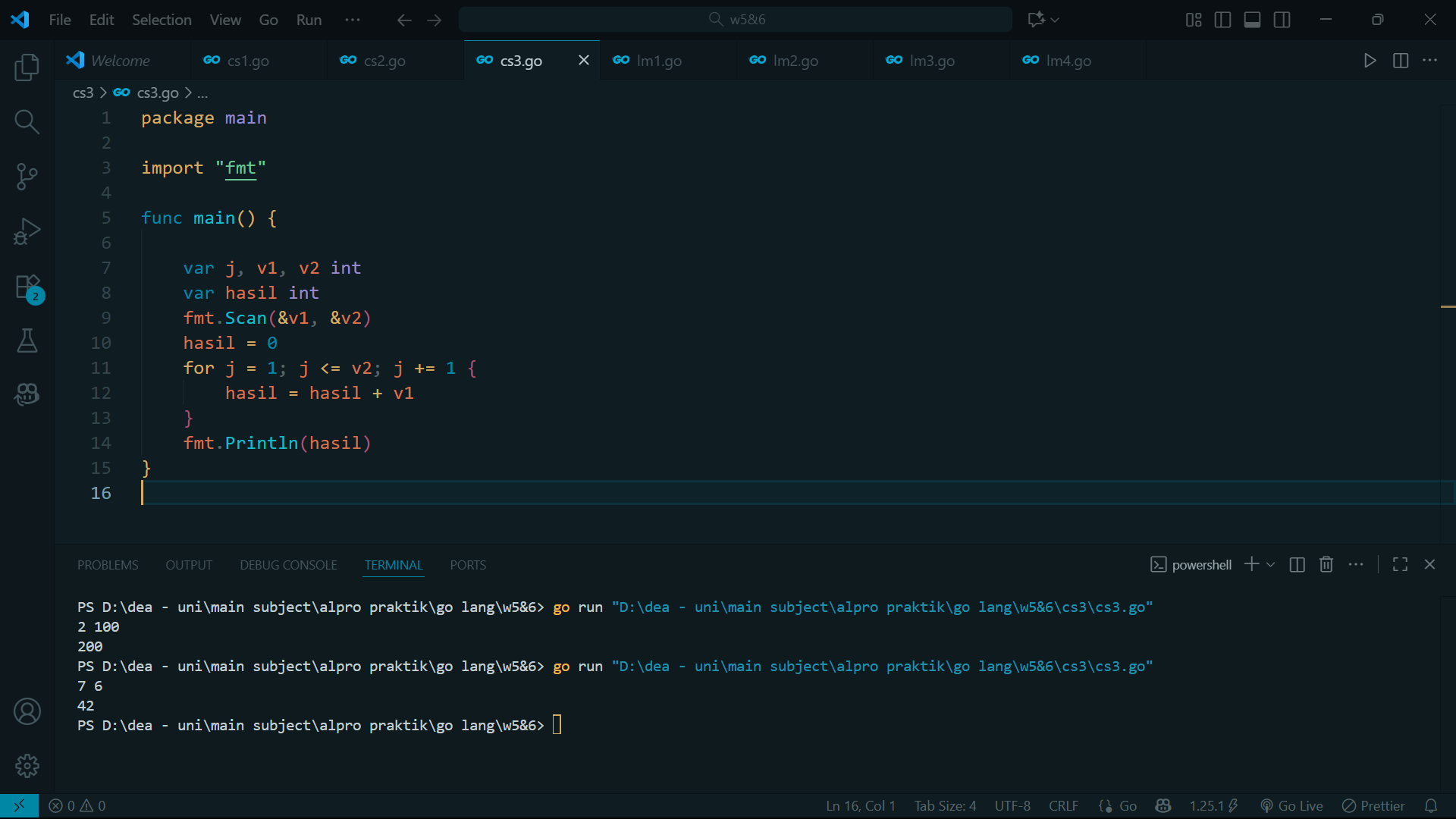
* + - 1. Menunggu user melakukan input data (fmt.Scan(&n)) yang nantinya data tersebut akan dimasukkan ke dalam variabel n. Hasil dari kode di baris ini adalah 5
      2. Melakukan perulangan variabel j (for j = 1; j <= n; j += 1) dengan nilai awal 1 (j = 1) dan setiap *command* yang ada dalam kurung kurawal setelah kode baris perulangan ini akan dilakukan berulang kali sebanyak hingga nilai variabel j menyentuh kurang dari sama dengan nilai variabel n, yaitu 5.
      3. Menunggu user melakukan input data (fmt.Scan(&alat, &tinggi)) dan nantinya nilai yang diinput tersebut akan dimasukkan ke dalam variabel alas dan variabel tinggi sesuai urutan input dari user. Setiap 2 nilai yang diinput oleh user terhitung 1 karena scan data variabel alas dan variabel tinggi terdapat dalam 1 kode baris yang sama. Oleh karena itu, input nilai terulang sebanyak 5 kali sesuai nilai variabel n, yaitu 5. Hasil dari kode di baris ini adalah 11 2 32 14 6 2 15 15 20 35
      4. Melakukan operasi hitung untuk mencari nilai luas segitiga dengan menggunakan rumus luas segitiga (luas = 0.5 \* float64(alas\*tinggi)) dan nilai yang dioperasikan menggunakan nilai dari data variabel alas dan variabel tinggi sebelumnya sesuai urutan input. Hasil dari kode di baris ini adalah operasi 0.5\*(11\*2) = 11, operasi dilakukan sesuai urutan input dan akan terus berulang hingga berhenti di perulangan operasi nilai ke-5.
      5. Melakukan print output dari variabel luas di baris baru (fmt.Println(luas)), yang mana output dari variabel luas didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah 11 224 6 112.5 350

1. **Guided 3**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var j, v1, v2 int      var hasil int      fmt.Scan(&v1, &v2)      hasil = 0      for j = 1; j <= v2; j += 1 {          hasil = hasil + v1      }      fmt.Println(hasil)  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Pertambahan Sebanyak v2 Kali** yang mana terlihat jelas melalui kode dari **baris 11 hingga baris 12**.

for j = 1; j <= v2; j += 1 {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for j = 1; j <= v2; j += 1

hasil = hasil + v1 🡪 Instruksi untuk melakukan operasi penambahan nilai variabel hasil (0) dengan nilai data variabel v1 (7) sebanyak nilai variabel v2 (6) yang nantinya akan dimasukkan ke dalam data variabel hasil dan operasi ini akan terus mengubah nilai variabel hasil itu sendiri

Runtutan Eksekusi:

* + - 1. Menunggu user melakukan input data (fmt.Scan(&v1, v2)) yang nantinya data tersebut akan dimasukkan ke dalam variabel v1 dan variabel v2 sesuai urutan input yang dilakukan oleh user. Hasil dari kode di baris ini adalah 7 6
      2. Menyatakan bahwa nilai variabel hasil adalah 0 (hasil = 0)
      3. Melakukan operasi perulangan (for j = 1; j <= v2; j += 1) yang mana nilai awal variabel j adalah sebanyak 1 dan perulangan dari kode baris ini akan terus diulang dan akan menambahkan setiap input atau nilai terbaru dengan nilai 1.
      4. Melakukan operasi hitung (hasil = hasil + v1) yang merupakan operasi di dalam kurung kurawal operasi perulangan di kode baris sebelumnya, sehingga operasi yang akan terus diulang adalah operasi hasil untuk mewakili variabel j dalam operasi for dan dilakukan sebanyak kurang dari sama dengan nilai variabel v2, yaitu 6. Setiap nilai terbaru yang dihasilkan oleh operasi ini akan dimasukkan ke dalam variabel hasil. Hasil dari kode di baris ini adalah operasi hasil (0) + v1 (7) = 7, nilai 7 akan masuk ke dalam variabel hasil yang terbaru, sehingga untuk perulangan kedua kali adalah sebagai berikut, hasil (7) + v1 (7) = 14, nilai 14 akan masuk ke dalam variabel hasil yang terbaru dan hasil terbaru dari operasi ini akan digunakan untuk melanjutkan perulangan hingga sebanyak nilai variabel v2, yaitu 6.
      5. Melakukan print output dari variabel hasil di baris baru (fmt.Println(hasil)), yang mana output dari variabel hasil didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah 42

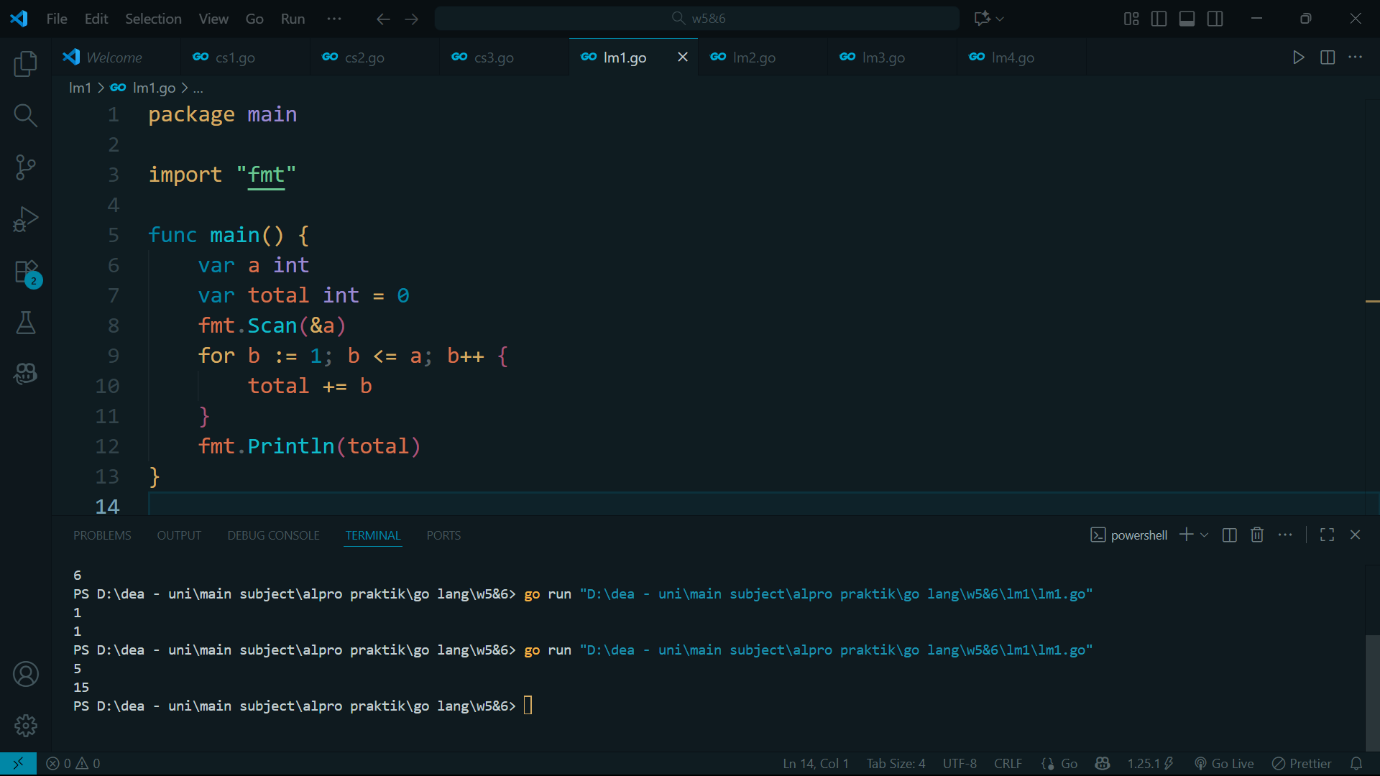
**TUGAS**

1. **Tugas 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var a int      var total int = 0      fmt.Scan(&a)      for b := 1; b <= a; b++ {          total += b      }      fmt.Println(total)  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Pertambahan Sebanyak Dari 1 Hingga a Kali** yang mana terlihat jelas melalui kode dari **baris 9 hingga baris 10**.

for b := 1; b <= a; b++ {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for b := 1; b <= a; b++

total += b 🡪 Instruksi untuk melakukan operasi penambahan sebanyak nilai variabel a (5) atau penambahan setiap angka yang terdapat dalam rentang nilai variabel a (5) dengan menggunakan nilai awal dari variabel b (1) dan untuk setiap perulangan dilakukan increment terhadap nilai variabel b (1) sebanyak 1, sehingga perulangan operasi hitung dalam rentang nilai variabel a (5) yang dimulai dengan nilai variabel b (1) mempunyai tampilan 1+2+3+4+5, yang mana di perulangan ke-2 didapatkan angka 2 karena nilai variabel b (1) telah dilakukan increment +1, sehingga 1+1 = 2, dan hal ini dilakukan untuk setiap perulangan

Runtutan Eksekusi:

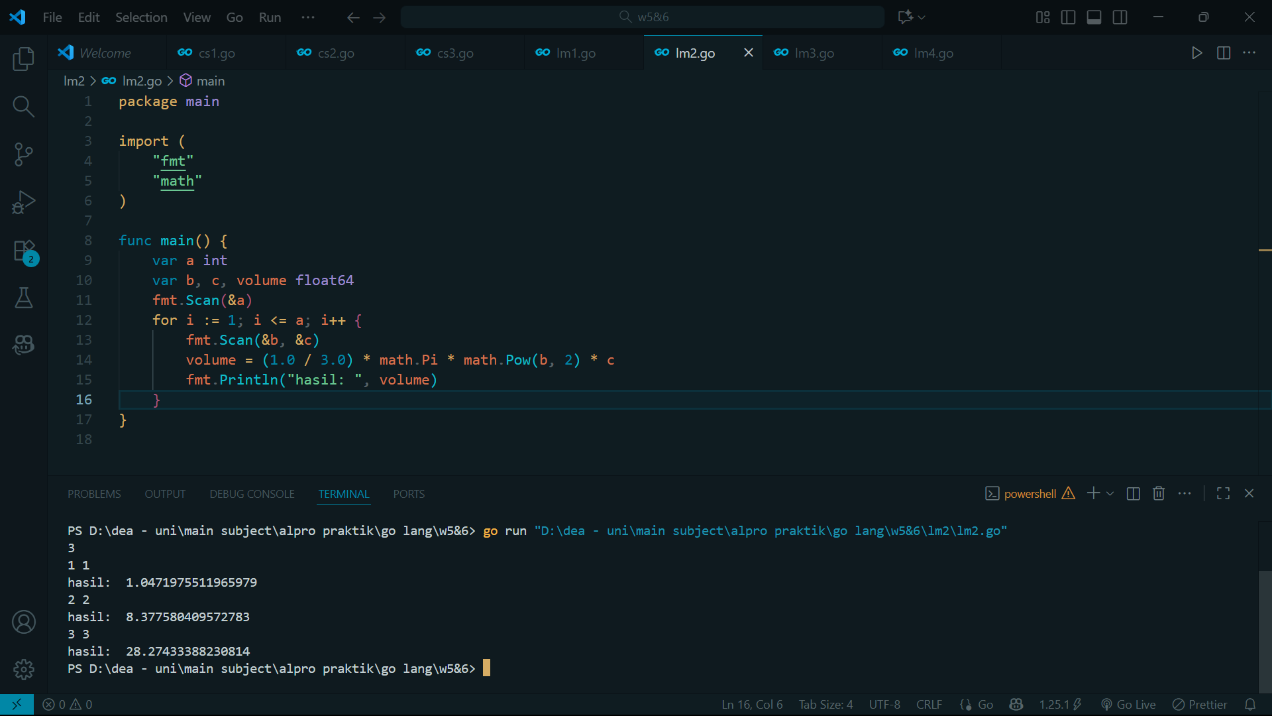
* + - 1. Menunggu user melakukan input data (fmt.Scan(&a)) yang nantinya data tersebut akan dimasukkan ke dalam variabel a. Hasil dari kode di baris ini adalah 5
      2. Melakukan operasi perulangan (for b := 1; b <= a; b++) yang mana nilai awal variabel b adalah sebanyak 1 dan perulangan dari kode baris ini akan terus diulang, dari setiap perulangan yang dilakukan sebanyak kurang dari sama dengan variabel a, maka variabel b akan dilakukan increment di setiap perulangan selanjutnya.
      3. Melakukan operasi hitung (total += b) di mana akan dimulai dari nilai 1, yaitu nilai variabel b dan nilai 1 tersebut akan terkena increment dari (+= b), sehingga di setiap perulangan nilai b akan terus ditambah dengan nilai 1 dan nilai dari variabel b di setiap perulangan akan selalu berubah. Hasil dari kode di baris ini adalah operasi penjumlahan dan increment dari variabel b sebanyak variabel a, dilakukan dengan cara 1+2+3+4+5 = 15
      4. Melakukan print output dari variabel total di baris baru (fmt.Println(total)), yang mana output dari variabel total didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah 15

1. **Tugas 2**

**Source code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var a int      var b, c, volume float64      fmt.Scan(&a)      for i := 1; i <= a; i++ {          fmt.Scan(&b, &c)          volume = (1.0 / 3.0) \* math.Pi \* math.Pow(b, 2) \* c          fmt.Println("hasil: ", volume)      }  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Terdapat "math" dengan peran sebagai library yang memiliki fungsi matematika lengkap di dalamnya, menyediakan berbagai fungsi dan konstanta matematika untuk membantu menjalankan program.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Volume Sejumlah a Kerucut** yang mana terlihat jelas melalui kode dari **baris 12 hingga baris 14**.

for i := 1; i <= a; i++ {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for i := 1; i <= a; i++

fmt.Scan(&b, &c) 🡪 Instruksi untuk menunggu user melakukan input data sebanyak 2 nilai yang akan dimasukkan ke dalam variabel b dan variabel c sesuai urutan input dari user

volume = (1.0 / 3.0) \* math.Pi \* math.Pow(b, 2) \* c 🡪 Instruksi untuk melakukan operasi hitung volume dengan rumus (1.0 / 3.0) \* math.Pi \* math.Pow(b, 2) \* c yang mana di setiap huruf b dan c merupakan nilai variabel b dan variabel c sesuai urutan input dari user. Hasil dari (1.0 / 3.0) atau sepertiga akan dikalikan dengan nilai math.Pi (π atau 3.14..se-persekian) dan math.Pow (mewakili simbol pangkat) (b, 2 -artinya b2 atau nilai variabel b pangkat 2) kemudian dikalikan dengan nilai dari variabel c.

Runtutan Eksekusi:

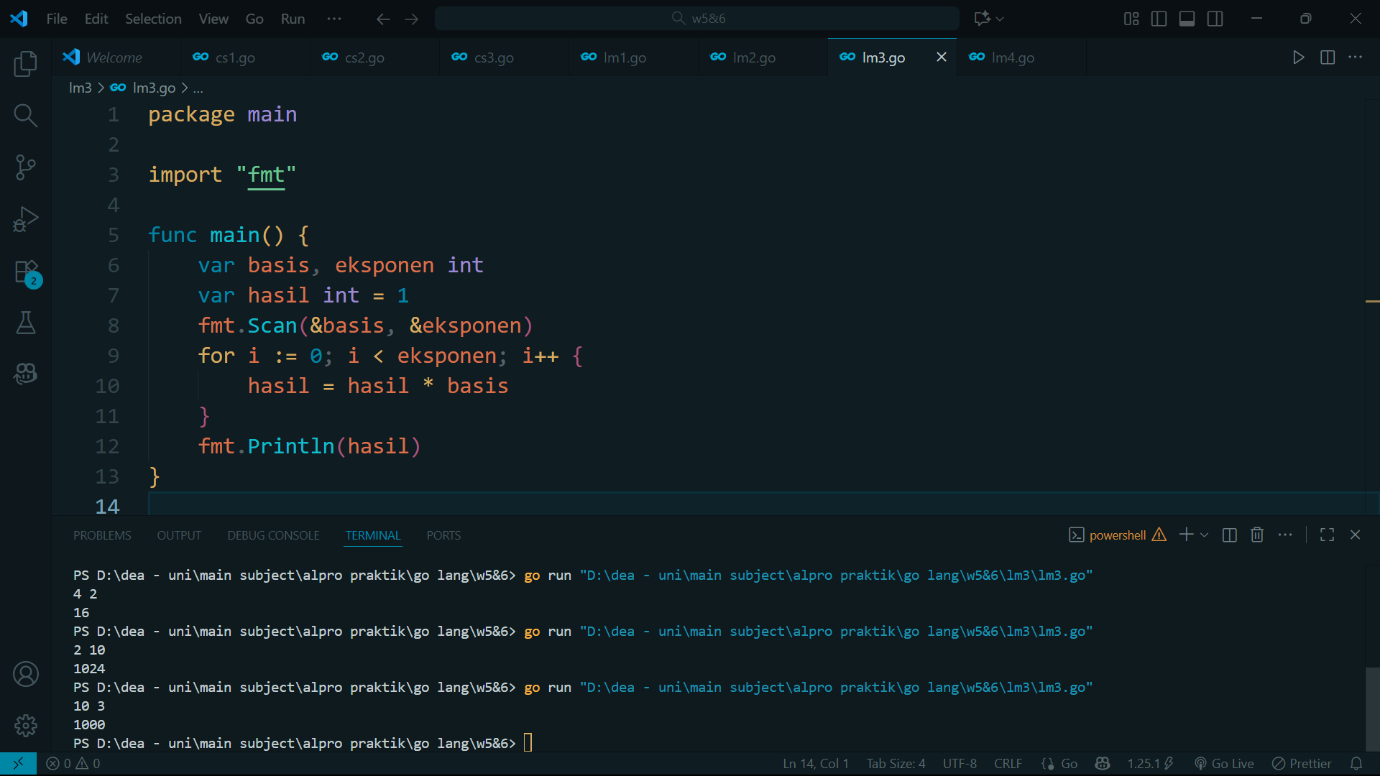
* + - 1. Menunggu user melakukan input data (fmt.Scan(&a)) yang nantinya data tersebut akan dimasukkan ke dalam variabel a. Hasil dari kode di baris ini adalah 3. Artinya perulangan akan dilakukan sebanyak 3 kali dan input data akan dilakukan sebanyak 3 kali.
      2. Melakukan operasi perulangan (for i := 1; i <= a; i++) yang mana nilai awal variabel i adalah sebanyak 1 dan perulangan dari kode baris ini akan terus diulang, dari setiap perulangan yang dilakukan sebanyak kurang dari sama dengan variabel a (3), maka variabel i akan dilakukan increment di setiap perulangan selanjutnya.
      3. Menunggu user melakukan input data (fmt.Scan(&b, &c)) yang nantinya data tersebut akan dimasukkan ke dalam variabel b dan variabel c sesuai urutan input dari user. Hasil dari kode di baris ini adalah 1 1. Artinya perulangan yang akan dilakukan untuk perulangan pertama menggunakan nilai data variabel b dan variabel c, yaitu 1 dan 1.
      4. Melakukan operasi hitung volume (volume = (1.0 / 3.0) \* math.Pi \* math.Pow(b,2) \* c) di mana operasi ini akan dimulai dari operasi pembagian dalam kurung (1.0 / 3.0) dengan hasil (0.333... se-persekian), dilanjutkan dengan mengalikan hasil tersebut dengan nilai math.Pi (3.14... se-persekian) yang menghasilkan 1.0472, lalu hasil tersebut dikalikan dengan math.Pow (12) dan akan dikalikan dengan nilai variabel c, yaitu 1.
      5. Melakukan print output dari variabel volume di baris baru (fmt.Println(volume)), yang mana output dari variabel volume didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah hasil: 1.0471975511965979
      6. Operasi ini akan tetap berjalan hingga perulangan menyentuh angka kurang dari sama dengan nilai variabel a, yaitu 3.

1. **Tugas 3**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var basis, eksponen int      var hasil int = 1      fmt.Scan(&basis, &eksponen)      for i := 0; i < eksponen; i++ {          hasil = hasil \* basis      }      fmt.Println(hasil)  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Pangkat Dua Bilangan Sebanyak Dari 1 Hingga n Kali** yang mana terlihat jelas melalui kode dari **baris 9 hingga baris 10**.

for i := 0; i < eksponen; i++ {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for i := 0; i < eksponen; i++ . Dalam perulangan ini dapat diartikan bahwa perulangan variabel i dimulai dari nilai 0 yang mana terulang sebanyak kurang dari nilai variabel eksponen (2) dan setiap perulangan nilai i akan terkena increment. Karena nilai i dimulai dari angka 0 dan nilai variabel eksponen adalah 2, berarti perulangan dilakukan sebanyak dari 0 hingga 1, yang mana nilai 1 kurang dari 2.

hasil = hasil \* basis 🡪 Instruksi untuk melakukan operasi perkalian untuk mendapatkan nilai pangkat tanpa langsung memangkatkan sebanyak nilai variabel eksponen (2)

Runtutan Eksekusi:

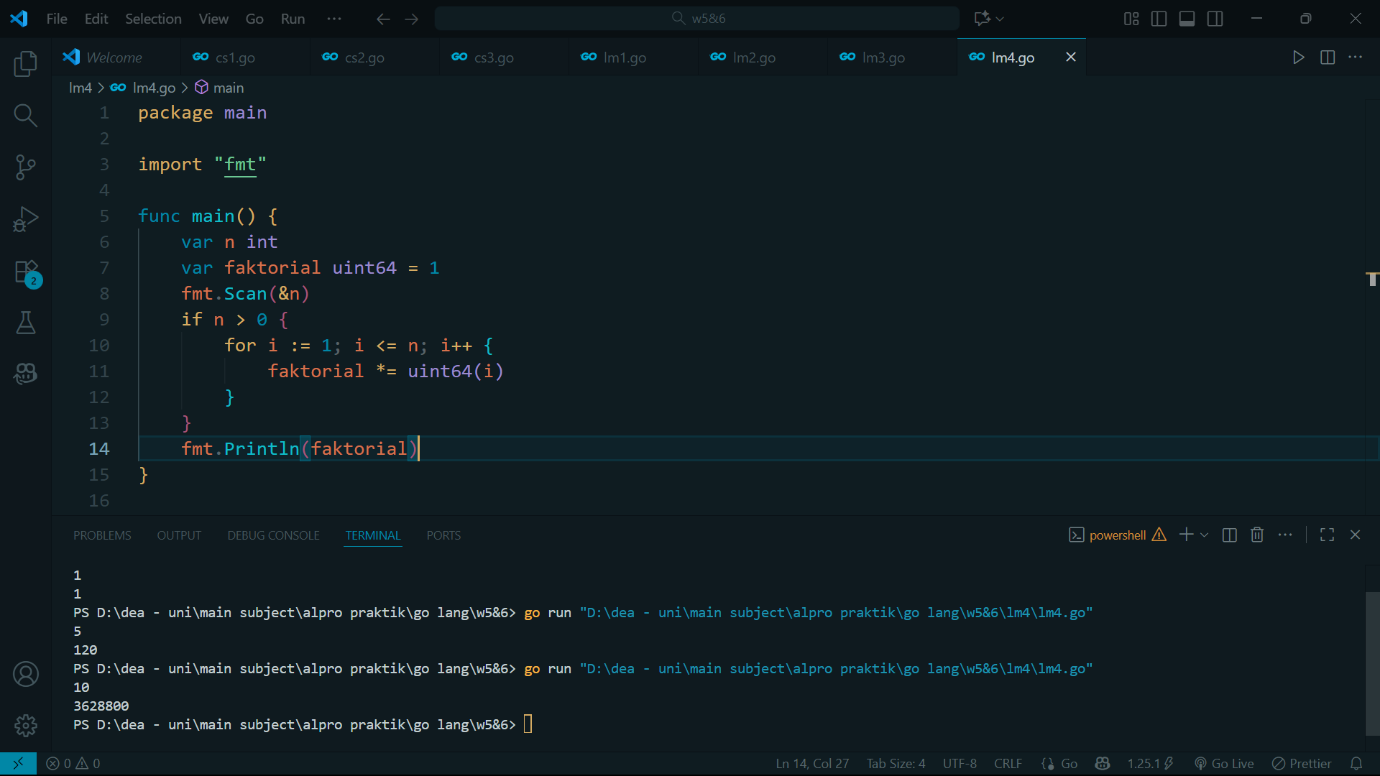
* + - 1. Menunggu user melakukan input data (fmt.Scan(&basis, &eksponen)) yang nantinya data tersebut akan dimasukkan ke dalam variabel basis dan variabel eksponen sesuai urutan input user. Hasil dari kode di baris ini adalah 4 2
      2. Melakukan operasi perulangan (for i := 0; i < eksponen; i++) yang mana nilai awal variabel i adalah sebanyak 0 dan perulangan dari kode baris ini akan terus diulang sebanyak kurang dari nilai variabel eksponen (2), dari setiap perulangan yang dilakukan sebanyak kurang dari variabel eksponen, maka variabel i akan dilakukan increment di setiap perulangan selanjutnya.
      3. Melakukan operasi hitung (hasil = hasil \* basis) di mana akan dimulai dari nilai hasil, yaitu 1 yang mana telah dideklarasikan di kode baris 7 (var hasil int = 1). Operasi dilakukan dengan perulangan ke-0 dengan mengalikan nilai hasil (1) dengan nilai basis yang diinput oleh user (4) dan menghasilkan nilai 4 yang mana nilai 4 tersebut akan masuk ke variabel hasil dan menjadi nilai terbaru dari variabel hasil untuk digunakan dalam operasi perulangan selanjutnya. Perulangan ke-1 dilakukan dengan menggunakan nilai hasil yang terbaru, yaitu 4, dikalikan dengan nilai variabel basis, yaitu 4, yang menghasilkan nilai 16 dan nilai 16 merupakan hasil pangkat dari 42.
      4. Melakukan print output dari variabel hasil di baris baru (fmt.Println(hasil)), yang mana output dari variabel hasil didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah 16

1. **Tugas 4**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var n int      var faktorial uint64 = 1      fmt.Scan(&n)      if n > 0 {          for i := 1; i <= n; i++ {              faktorial \*= uint64(i)          }      }      fmt.Println(faktorial)  } |

**Screenshoot program**



**Deskripsi program**

Permulaan program ini dimulai denganpackage main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

Dilanjutkan dengan import “fmt” yang digunakan untuk memasukkan library “fmt” untuk dipakai menjalankan program nantinya.

Lalu, terdapat func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

Fungsi var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

Terdapat fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di line baru.

Kemudian fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.  
Ada pula if yang berfungsi untuk memeriksa suatu kondisi dengan hasil true or false ataupun runtutan alur yang jika sesuai dengan suatu kondisi yang sesuai akan dilanjutkan ke perintah selanjutnya atau tidak.

Lalu for berfungsi sebagai perulangan, yang merupakan proses mengulang-ulang eksekusi blok kode di dalam perintah for tanpa henti selama kondisi yang diperintahkan terpenuhi, dalam blok kode for juga biasanya terdapat variabel penanda kapan perulangan akan diberhentikan.

Yang dilakukan program di dalam gambar di atas menurut saya adalah **Perulangan Operasi Hitung Faktorial Suatu Bilangan Sebanyak Dari n Kali** yang mana terlihat jelas melalui kode dari **baris 9 hingga baris 11**.

if n > 0 {} 🡪 Instruksi untuk melakukan pengecekan kondisi dari variabel n memenuhi syarat lebih dari 0 atau tidak, yang mana jika memenuhi syarat melebihi 0, maka dilanjutkan ke program for atau perulangan karena terdapat dalam kurung kurawal program if atau merupakan runtutan dari program if sebelumnya

for i := 1; i <= n; i++ {} 🡪 Memberikan instruksi untuk melakukan perulangan *command* yang terdapat dalam kurung kurawal setelah kode baris for i := 1; i <= n; i++ . Dalam perulangan ini dapat diartikan bahwa perulangan variabel i dimulai dari nilai 1 yang mana terulang sebanyak kurang dari sama dengan nilai variabel n (5) dan setiap perulangan nilai i akan terkena increment. Karena nilai i dimulai dari angka 1 dan nilai variabel n adalah 5, berarti perulangan dilakukan sebanyak dari 1 hingga 5, yang mana nilai 5 sama dengan nilai 2.

Faktorial \*= uint(i) 🡪 Instruksi untuk melakukan operasi perkalian untuk mendapatkan nilai faktorial dari variabel n

Runtutan Eksekusi:

* + - 1. Menunggu user melakukan input data (fmt.Scan(&n)) yang nantinya data tersebut akan dimasukkan ke dalam variabel n. Hasil dari kode di baris ini adalah 5
      2. Melakukan pengecekan kondisi (if n > 0) apakah nilai variabel n lebih dari 0 atau tidak, jika iya, maka akan dilanjutkan ke program selanjutnya. Karena nilai variabel n yang diinput oleh user adalah 5 dan nilai 5 lebih besar dari 0, maka dilanjutkan ke program selanjutnya
      3. Melakukan operasi perulangan (for i := 1; i <= n; i++) yang mana nilai awal variabel i adalah sebanyak 1 dan perulangan dari kode baris ini akan terus diulang sebanyak kurang dari sama dengan nilai variabel n (5), dari setiap perulangan yang dilakukan sebanyak kurang dari sama dengan variabel n, maka variabel n akan dilakukan increment di setiap perulangan selanjutnya.
      4. Melakukan operasi hitung (faktorial \*= uint64(i)) di mana proses hitung di kode baris ini dimulai dari angka yang masuk ke dalam tipe data uint64, karena i dimulai dari angka 1, maka perkalian faktorial akan dimulai angka 1 yang dikalikan dengan angka 2 karena setiap perulangan lanjutan dikenakan increment (\*=), dari hasil perkalian 1\*2, didapatkan nilai 2 yang dilanjutkan untuk dikalikan dengan angka 3 dan seterusnya hingga mencapai perulangan nilai kurang dari sama dengan variabel n, yaitu 5.
      5. Melakukan print output dari variabel hasil di baris baru (fmt.Println(faktorial)), yang mana output dari variabel faktorial didapatkan dari perulangan operasi di kode baris sebelumnya. Hasil dari kode di baris ini adalah 120